Vysoká škola ekonomická

Analýza časových řad v prostředí R

=== 4ST431 Časové řady ===

=== semestrální práce ===

Lubomír Štěpánek

***

prosinec 2017

***

Obsah

Vysázeno pomocí R-ka, YAML-u, TEX-u a BibTEX-u

1 Úvod

Tato práce předkládá řešení semestrálního zadání v rámci předmětu 4ST431 Časové řady, vyučovaném na Vysoké škole ekonomické v Praze.

Smyslem je analyzovat jednu nesezónní a jednu sezónní časovou řadu, u každé řady správně určit model, diagnosticky ho zkontrolovat a rovněž ověřit dodržení předpokladů.

2 Zadání úlohy

Východiskem jsou dvě jednorozměrné časové řady, jedna nesezónní a druhá sezónní. Delší časové řady jsou preferovány. Cílem je u každé z řad vytvořit odhad modelu, tj. buďto ARIMA (AuroRegressive Integrated Moving Average), či SARIMA (Seasonal AuroRegressive Integrated Moving Average) model.

Postup je třeba podložit veškerými mezivýsledky a pokud možno diagramy a diagnostickými testy[1].

Zdrojem dat mohou být databáze Českého statistického úřadu (ČSÚ), Eurostatu, elektronických zdrojů VŠE apod.

Postup prací by měl sledovat

(i) zjištění, zda jde o nesezónní, či sezónní časovou řadu;

(ii) zjištění, zda jde o stacionární časovou řadu;

(iii) odhadnutí parametrů nutných k vystihnutí ARIMA, resp. SARIMA modelu;

(iv) diagnostiku odhadnutého modelu.

[1] Predikce pomocí modelů vytvořených nad každou řadou nejsou povinné.

3 Řešení úlohy

3.1 Metodologie a analýza dat

3.1.1 Použitá data a nástroje

Datový soubor nesezónní jednorozměrné časové řady pochází z portálu Epidemiologie zhoubných nádorů v České republice, dostupném na adrese

http://www.svod.cz/,

a jedná se o časovou řadu s každoročním záznamem incidence karcinomu tlustého střeva (tedy počet nových diagnóz na sto tisíc osob v riziku) na území České republiky, postupně mezi lety 1977 až 2015. Dataset je konkrétně dostupný na

http://www.svod.cz/analyse.php?modul=incmor#,

a je uložen v doprovodném souboru `karcinom_tlusteho_streva.txt`.

Datový soubor sezónní jednorozměrné časové řady pochází naopak z webu prof. Bolkera z Univerzity McMaster v Kanadě, věnovaném epidemiologii některých infekčních nemocí. Web je dostupný na adrese

https://ms.mcmaster.ca/ bolker/measdata.html.

Konkrétně byl použit dataset zaznamenávající každotýdenní počet nových případů spalniček a planých neštovic na území vybraných měst Anglie v letech 1948 až 1987. Dataset však před použitím vyžadoval preprocessing, rovněž byla aplikována agregace záznamů na čtvrtletní periodu. Pro samotné analýzy byla použita data pro město Londýn. Dataset je možné stáhnout přímo v linku

https://ms.mcmaster.ca/ bolker/measdata/ewcitmeas.dat

a je zároveň uložen v doprovodném souboru `ewcitmeas.dat`.

Celá úloha byla řešena v prostředí R, které je určeno pro statistické výpočty a následné grafické náhledy [1]. Jednotlivé operace s daty a výpočty v rámci analýzy časových řad byly provedeny pomocí R-kových balíčků `stats`, `xtable`, `openxlsx`, `tseries`, `seasonal`, `forecast`, `lmtest` a `car`.

3.1.2 Testování sezónnosti

Sezónnost byla otestována R-kovou implementací procedury Census X13, více v [2], která je založena na F-testu zkoumajícím nulovou hypotézu H_0 o shodnosti rozptylů ve všech podmnožinách záznamů časové řady rozčleněných do kategorií dle periody předpokládané sezónnosti (v našem případě 1. až 4. čtvrtletí), a to pomocí statistiky

$$F = \frac{1}{(N-K)} \sum_{i=1}^{K} \sum_{j=1}^{n_i} \left(Y_{ij} - \bar{Y}_{i\cdot} \right)^2 \sim F(K-1, N-K) \mid H_0,$$

kde Y_{ij} je j-té pozorování v i-té z K skupin (zde $K = 4$) a N je celkový počet pozorování. Zřejmě F statistika sleduje Fisherovo-Snedeckerovo rozdělení o stupních volnosti $(K - 1, N - K)$ za předpokladu nulové hypotézy.

3.1.3 Testování stacionarity časové řady

Časovou řadu považujeme za stacionární, pokud se v čase nemění střední hodnota a variabilita sledované jednorozměrné veličiny, [3]. Stacionaritu lze ověřovat pomocí augmentovaného Dickey-Fullerova testu, který testuje nulovou hypotézu H_0 o přítomnosti jednotkového kořenu ve vzorku časové řady (časová řada je nestacionární) proti alternativní hypotéze H_1 o jeho nepřítomnosti, tj. kdy je časová řada stacionární, více v [4].

Alternativou je pak Kwiatkowski-Phillips-Schmidt-Shinův (KPSS) test testující nulovou hypotézu o stacionaritě řady, [5].

3.1.4 Budování modelu ARIMA, resp. SARIMA

Model ARIMA (AuroRegressive Integrated Moving Average) pro jednorozměrnou časovou řadu y_t je dán rovnicí

$$\hat{y}_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-i} - \sum_{j=1}^{q} \theta_j e_{t-j},$$

kde $\forall i \in \{1, 2, \ldots, p\}$ jsou ϕ_j parametry autoregresních procesů a $\forall j \in \{1, 2, \ldots, q\}$ jsou θ_j parametry klouzavých průměrů, dále μ je konstanta, [6]. Parametry p a q je třeba vhodně odhadnout pro vystižení vzorku časové řady $\hat{y}_t$ tak, aby její předpis by co nejjednodušší a její rezidua splňovala podmínky tzv. bílého šumu (též tzv. slabou sadu předpokladů), tj. (i) $\mathbf{E}(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t) = \mathbf{0}$ (střední hodnota chybové složky je nulová), (ii) $\mathrm{cov}(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t) = \sigma^2 \boldsymbol{I}$ (homoskedasticita a absence autokorelace chybové složky) a nakonec $(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t) \sim \mathcal{N}(\mathbf{0}, \sigma^2)$ (chybová složka sleduje normální rozdělení), [6]. Funkčně se někdy konkrétní model ARIMA zapisuje též jako ARIMA(p, d, q), kde p a q odpovídá počtu autoregresních členů a počtu členů zpoždění v rovnici, respektive (viz výše), a d je počet nutných diferencování časové řady k získání její stacionarity.

Obdobně model SARIMA (Seasonal AuroRegressive Integrated Moving Average) procesu X_t má tvar $SARIMA(p, d, q)(P, D, Q)_m$ tak, že

$$\Phi(B^m)\phi(B)\nabla_m^D \nabla^d X_t = \Theta(B^m)\theta(B)Z_t,$$

kde Z_t je proces bílého šumu, m je délka sezónní periody a B je tzv. *backshift* operátor; ostatní značení zůstává shodné s ARIMA modelem, velká písmena (P, D, Q) značí obdobné parametry, ale sezónních složek modelu.

Pro orientační odhady parametrů (p, d, q) a (P, D, Q) mohou dobře sloužit korelogramy, tj. diagram autokorelační (ACF) a parciální autokorelační (PACF) funkce. Autokorelační funkce (ACF) má těsnější vztah k procesu klouzavých průměrů, zatímco parciální autokorelační funkce (PACF) zase

k autoregresnímu procesu. Nepravidelnosti či nežádoucí periodicity v korelogramech tak lze použít o odhadu parametrů p a q.

Pro komplikovanější průběhy korelogramů někdy není snadné nahlédnout, čemu by se měly parametry p, q (a d) rovnat, pak je možné exhaustivně prozkoumat všechny modely ARIMA(i, d, j) pro $\forall i \leq p$ a $\forall j \leq q$, kde p a q jsou dostatečně nadhodnoceny, a vybrat takový model ARIMA(i^*, d, j^*), že např. Akaikeho informační kritérium (AIC) či Bayesovo informační kritérium (BIC) je pro model ARIMA(i^*, d, j^*) nejmenší možné. V R je pro exhaustivní prozkoumávání modelů podle minimalizace AIC připravena funkce `auto.arima()`, která vrátí parametry (p, d, q), resp. $(P, D, Q)_m$ nejinformativnějšího modelu (dle AIC či BIC).

3.1.5 Ověřování absence autokorelace chybové složky

Absenci autokorelace chybové složky je možné ověřit pomocí Breusch-Godfreyho testu seriálových korelací, který testuje nulovou hypotézu H_0 o nepřítomnosti autokorelace chybové složky v časové řadě, [7].

3.1.6 Ověřování normality chybové složky

Pro ověřování jednorozměrné normality dat spojitých proměnných byly použity histogramy, dále Shapirův-Wilkův test. Ten testuje nulovou hypotézu H_0 o tom, že statistický výběr $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$ pochází z normálního rozložení. Testová statistika je

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

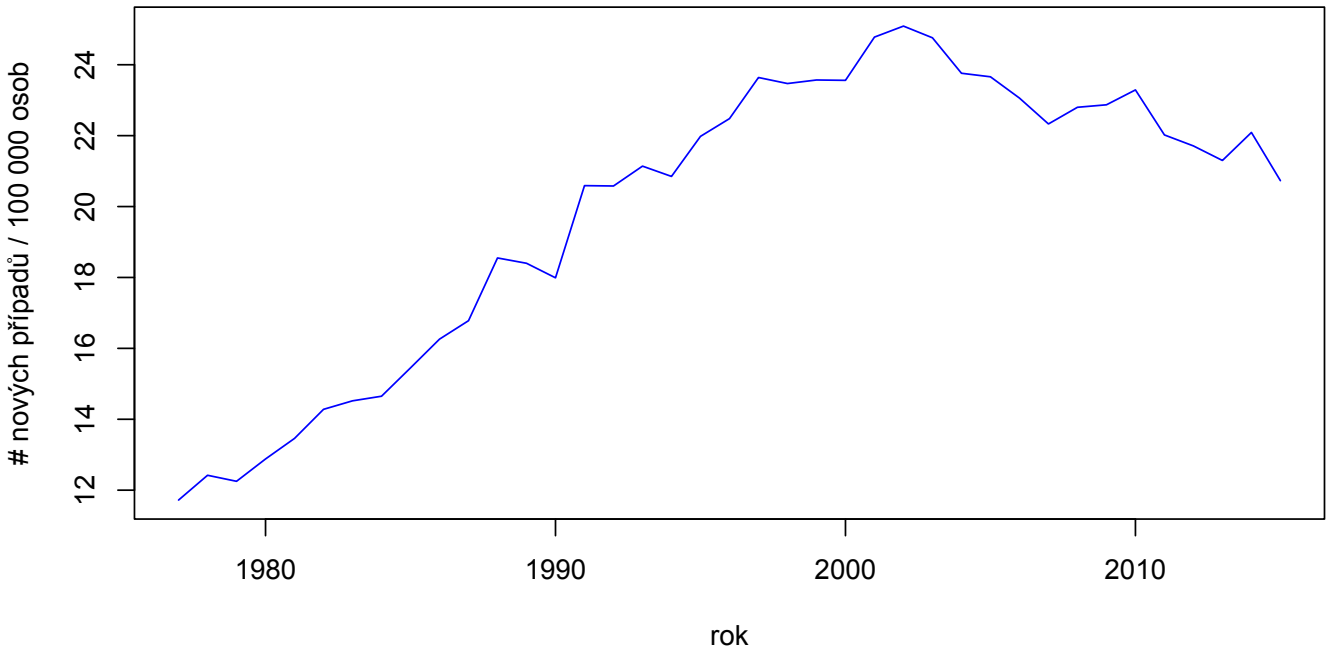
kde $x_{(i)}$ je i-tá nejmenší hodnota výběru $\boldsymbol{x}$, $\bar{x}$ je výběrový průměr a a_i jsou konstanty dány Shapiro-Wilkovou metodikou podle [8]. Kritické hodnoty pro statistiku W jsou pro danou hladinu významnosti α tabelovány nebo dostupné ve vhodném software. V prostředí R je Shapirův-Wilkův test implementován ve funkci `shapiro.test()`.

3.1.7 Ověřování homoskedasticity chybové složky

Breusch-Paganův test testuje nulovou hypotézu H_0 o homoskedasticitě chybové složky, [9].

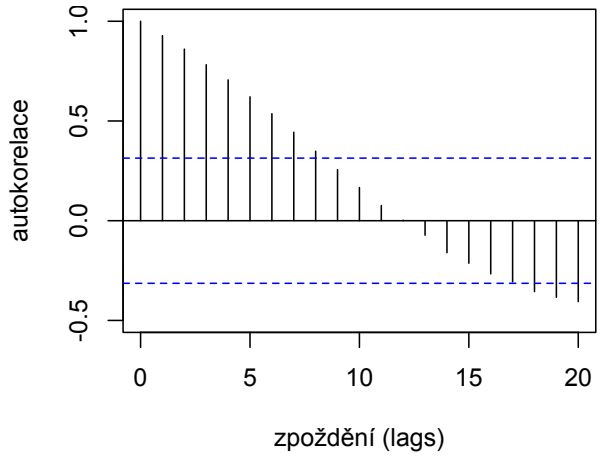
3.2 Výsledky pro nesezónní časovou řadu

Nesezónní časová řada zkoumá incidenci nových případů karcinomu tlustého střeva v letech 1977 až 2015 na území České republiky. Z logiky věci nepředpokládáme sezónnost. Průběh nesezónní časové řady je na obrázku 1.

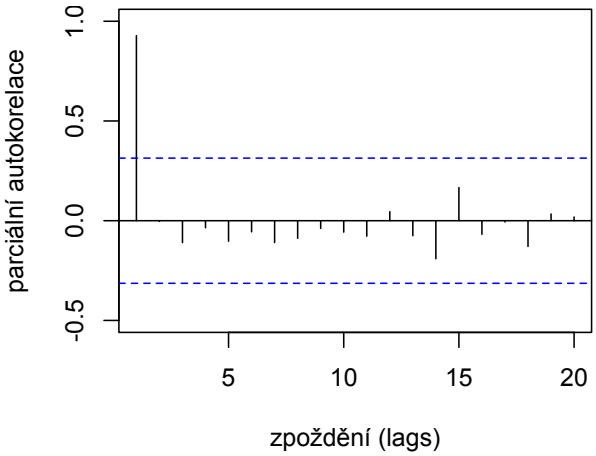


Obrázek 1: Průběh nesezónní časové řady

Na obrázcích 2 a 3 vidíme korelogramy autokorelační (ACF) a parciální autokorelační (PACF) funkce nad původními hodnotami nesezónní časové řady. Zřejmě lze již z grafického náhledu očekávat nestacionaritu.



Obrázek 2: Korelogram ACF původních hodnot nesezónní řady



Obrázek 3: Korelogram PACF původních hodnot nesezónní řady

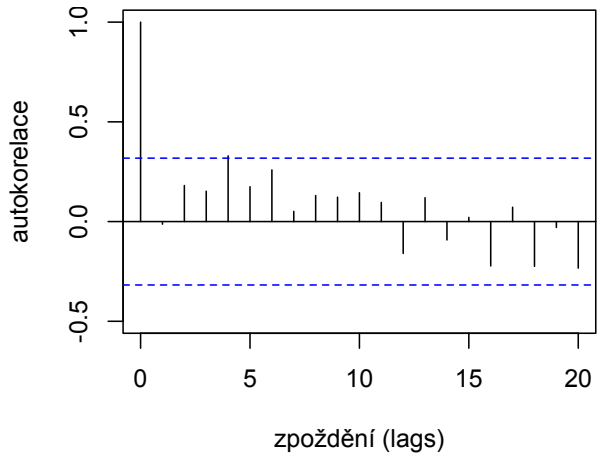
3.2.1 Test stacionarity časové řady

Byl proveden formální test nestacionarity nesezónní časové řady, a sice augmentovaný Dickey-Fullerův test.

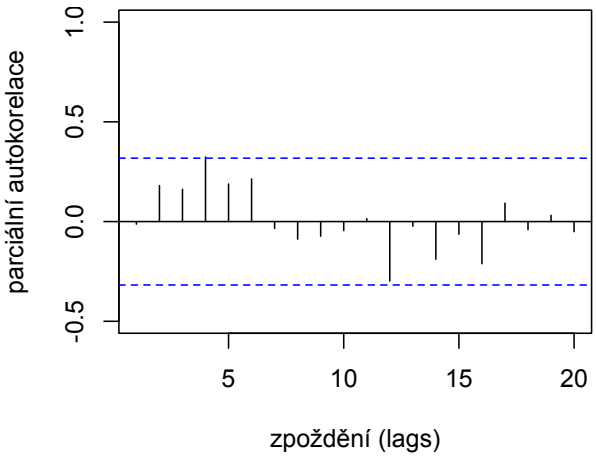
Pro původní data vrátil test hodnotu testové statistiky $DF = 0{,}734$ a hladinu významnosti $p = 0{,}990$. Vzhledem k nestacionaritě řady se nabízí její logaritmování, posléze spočítání prvních diferencí. Pro logaritmovanou řadu vrací augmentovaný Dickey-Fullerův test statistiku $DF = 0{,}369$ a hladinu významnosti $p = 0{,}990$. Až pro první diference logaritmované řady dostáváme stacionární průběh, kdy $DF = -3{,}829$ a hladina významnosti $p = 0{,}029$.

3.2.2 Korelogram upravené nesezónní časové řady

Na obrázcích 4 a 5 lze nahlédnou korelogram nad prvními diferencemi logaritmované nesezónní časové řady.



Obrázek 4: Korelogram ACF prvních diferencí logaritmů nesezónní řady



Obrázek 5: Korelogram PACF prvních diferencí logaritmů nesezónní řady

Z korelogramu ACF můžeme nahlédnou, že čtvrtý sloupeček je ještě hraničně u doporučené meze, pro autoregresní složku AR(p) modelu ARIMA tedy bude nejspíše platit $p \leq 4$. Naopak z korelogramu PACF a rovněž čtvrtého sloupečku korelace u meze se lze dovtípit, že pro složku klouzavých průměrů MA(q) bude nejspíš opět platit $q \leq 4$. Že je $d = 1$, víme již z testování stacionarity (bylo třeba jedné iterace diferencování, abychom dostali stacionární řadu, proto $d = 1$).

3.2.3 Určení nejvhodnějšího modelu

Protože se efekty složek AR(p) a MA(q) mohou v jednom modelu ARIMA navzájem „rušit“, nelze apriorně říci, že vhodným modelem bude ARIMA(4, 1, 4), ale je třeba vyzkoušet všech $|0, 1, \ldots, 4| \times |0, 1, \ldots, 4| = 25$ možností. Ty byly zkoušeny exhaustivně a v tabulce 1 je přehled modelů a některých jejich charakteristik.

Dle AIC nahlédneme, že nejinformativnějším modelem je ARIMA(2, 1, 2). V tabulce 2 je sumář modelu ARIMA(2, 1, 2); v sumáři není obsažena konstanta, byla tedy automaticky procedurou vyloučena jako nesignifikantní.

	p	d	q	AIC	Ljung-Box p	Breusch-Godfrey p	Shapiro-Wilk p	Breusch-Pagan p
1	0	1	0	-129,392	0,040	0,888	0,243	0,631
2	0	1	1	-127,745	0,034	0,479	0,189	0,652
3	0	1	2	-127,422	0,200	0,638	0,117	0,695
4	0	1	3	-126,140	0,093	0,741	0,412	0,845
5	0	1	4	-129,126	0,474	0,954	0,060	0,820
6	1	1	0	-127,943	0,038	0,314	0,157	0,664
7	1	1	1	-133,245	0,112	0,107	0,188	0,813
8	1	1	2	-125,109	0,224	0,550	0,046	0,560
9	1	1	3	-137,363	0,437	0,980	0,607	0,574
10	1	1	4	-136,278	0,399	0,914	0,565	0,748
11	2	1	0	-129,056	0,391	0,432	0,084	0,720
12	2	1	1	-134,239	0,422	0,729	0,638	0,947
13	2	1	2	-138,748	0,331	0,706	0,635	0,743
14	2	1	3	-137,615	0,531	0,966	0,888	0,713
15	2	1	4	-134,770	0,649	0,899	0,652	0,867
16	3	1	0	-128,825	0,126	0,433	0,495	0,989
17	3	1	1	-133,395	0,215	0,696	0,552	0,835
18	3	1	2	-136,993	0,394	0,863	0,827	0,777
19	3	1	3	-135,251	0,166	0,667	0,502	0,781
20	3	1	4	-133,907	0,795	0,687	0,739	0,771
21	4	1	0	-132,401	0,667	0,491	0,178	0,792
22	4	1	1	-134,391	0,723	0,771	0,240	0,929
23	4	1	2	-133,709	0,641	0,918	0,416	0,827
24	4	1	3	-133,220	0,382	0,902	0,767	0,871
25	4	1	4	-134,766	0,574	0,932	0,759	0,806

Tabulka 1: Modely ARIMA(i, 1, j) pro $\forall i \leq 4$ a $\forall j \leq 4$

	odhad	střední chyba	z-hodnota	p-hodnota
AR(1)	1,512	0,160	9,434	$< 0{,}001$
AR(2)	-0,537	0,159	-3,365	0,001
MA(1)	-1,874	0,196	-9,546	$< 0{,}001$
MA(2)	1,000	0,207	4,835	$< 0{,}001$

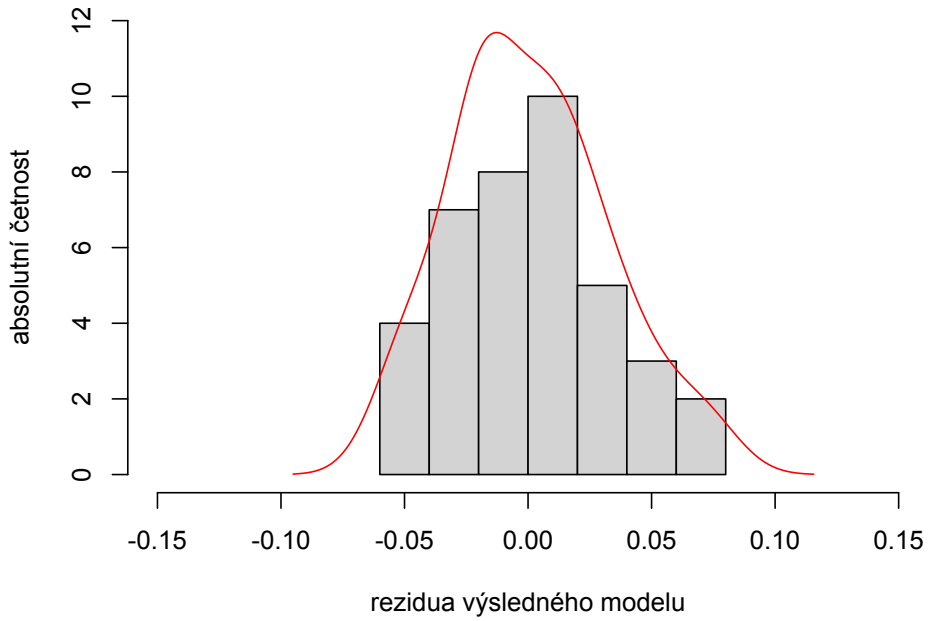
Tabulka 2: Sumář modelu ARIMA(2, 1, 2)

3.2.4 Diagnostika modelu

V tabulce 1 rovněž vidíme, že Ljung-Boxův test ($p = 0{,}331$) a Breusch-Godfreyho test ($p = 0{,}706$) nezamítá pro tento model nulovou hypotézu o absenci autokorelace, Shapiro-Wilkův test nezamítá

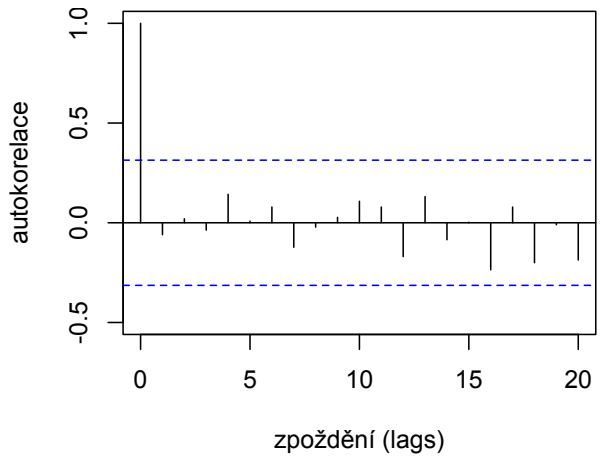
nulovou hypotézu o normalitě reziduí ($p = 0{,}635$) a Breusch-Paganův test nezamítá nulovou hypotézu o homoskedasticitě chybové složky ($p = 0{,}743$).

Dále ještě obrázek 6 ztvrzuje závěr Shapiro-Wilkova test, že rezidua modelu je možné považovat za normální.

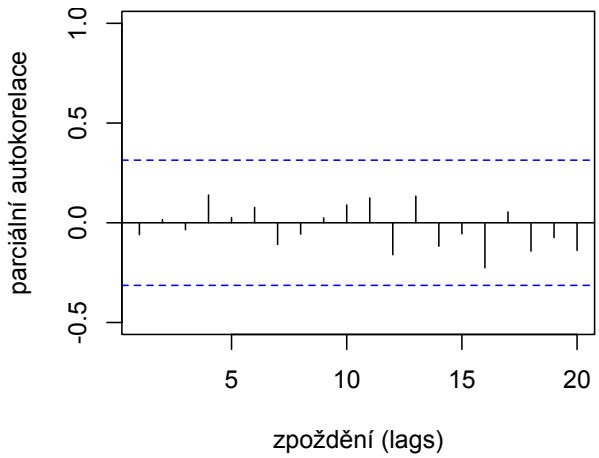


Obrázek 6: Histogram rozložení reziduí modelu ARIMA(2, 1, 2), červeně je naznačena křivka odhadu jádrové hustoty

Nakonec obrázky 7 a 8 naznačují chování reziduí modelu ARIMA(2, 1, 2). Není naznačena korelace ani jiná systematičnost, oba korelogramy jsou akceptovatelné.



Obrázek 7: Korelogram ACF reziduí modelu ARIMA(2, 1, 2) nad nesezónní řadou



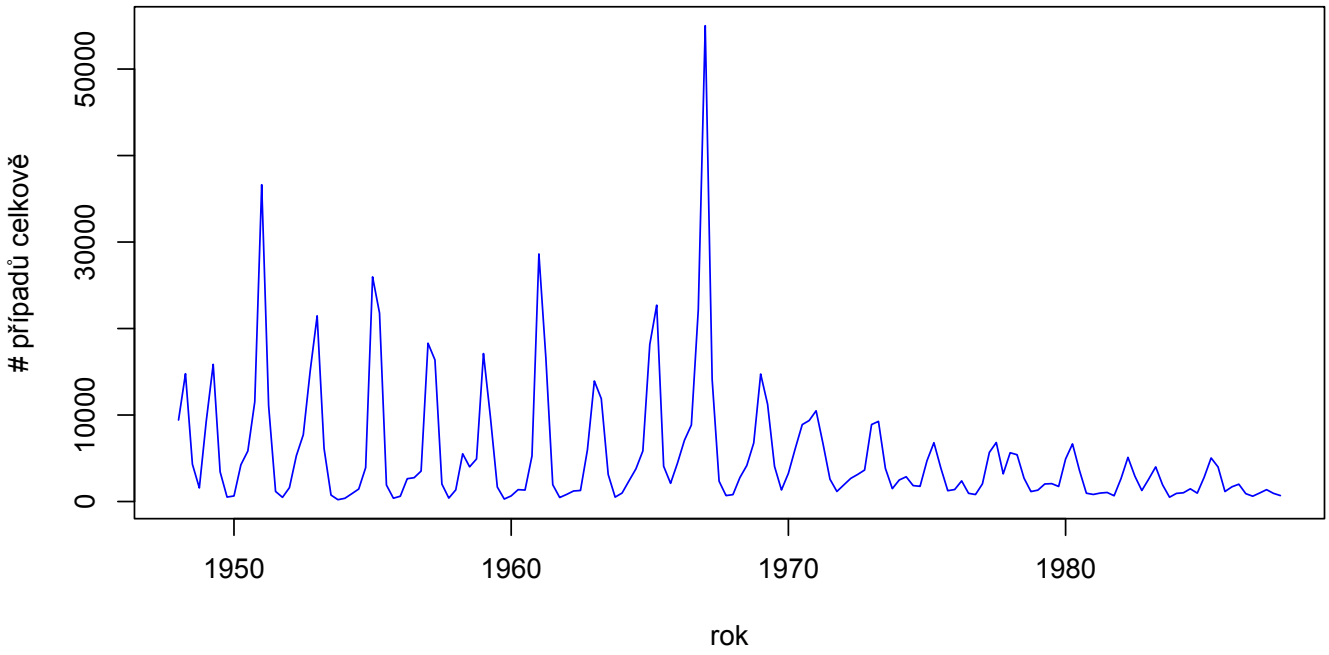
Obrázek 8: Korelogram PACF reziduí modelu ARIMA(2, 1, 2) nad nesezónní řadou

3.2.5 Závěr

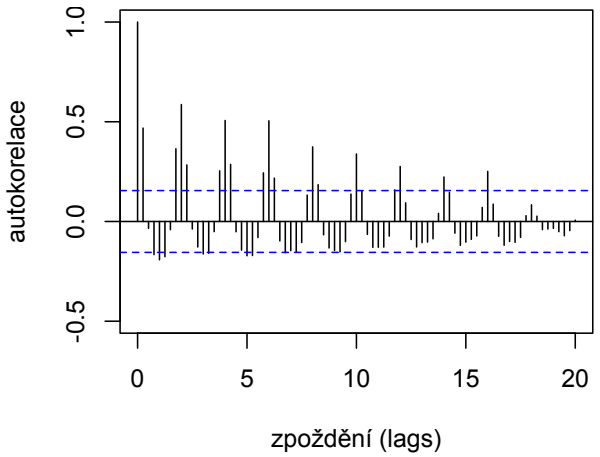
Pro nesezónní časovou řadu byl zvolen jako nejvýstižnější model ARIMA(2, 1, 2) bez konstanty.

3.3 Výsledky pro sezónní časovou řadu

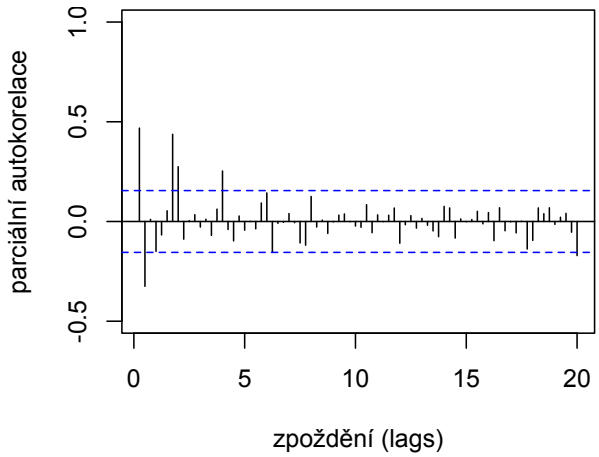
Sezónní časová řada zkoumá záchyt nových případů infekce spalniček či planých neštovic v Londýně v letech 1948 až 1987. Jde o jednorozměrnou časovou řadu se čtvrtletním úhrn záznamem nových případů. Z logiky věci lze sezónnost předpokládat, historické prameny odhadují periodický trend i sezónnost, více např. [10]. Průběh sezónní časové řady je na obrázku 9. Zajímavý je i fakt, že v Anglii bylo uvedeno povinné očkování proto spalničkám a planým neštovicím poprvé v roce 1968, od té doby lze vidět, že sezónnost se směrem k současnosti vyhlazuje, graficky se zdá, že ztrácí na významu; fenomén erupce (dříve typicky v dvouletých intervalech) spalniček a planých neštovic již není příliš výrazný.



Obrázek 9: Průběh sezónní časové řady



Obrázek 10: Korelogram ACF původních hodnot sezónní řady



Obrázek 11: Korelogram PACF původních hodnot sezónní řady

Na obrázcích 10 a 11 vidíme korelogramy autokorelační (ACF) a parciální autokorelační (PACF) funkce nad původními hodnotami sezónní časové řady. Zřejmě lze již z grafického náhledu očekávat nestacionaritu.

3.3.1 Test sezónnosti časové řady

Test sezónnosti časové řady byl proveden pomocí R-kové implementace procedury Census X13; jde v podstatě o F-test a jednocestnou analýzu rozptylu. Hladina významnosti F-testu pro sezónnost je $p < 0{,}001$, vyvrátili jsme tedy nulovou hypotézu H_0 o nesezónnosti řady.

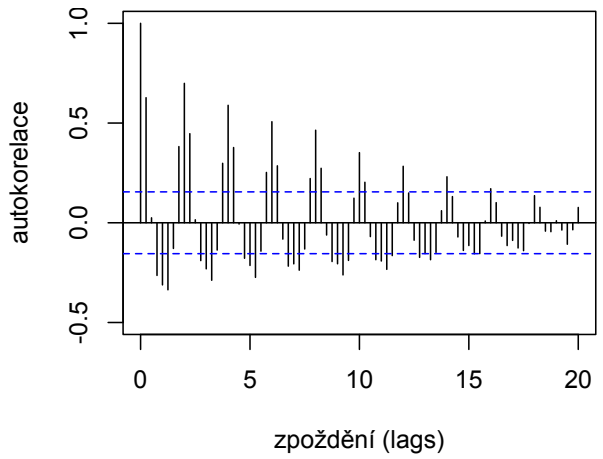
3.3.2 Test stacionarity časové řady

Byl proveden formální test nestacionarity sezónní časové řady, a sice augmentovaný Dickey-Fullerův test. Vzhledem k různě velkým magnitudám časové řady (s periodou cca dva roky) byly původní řady logaritmovány.

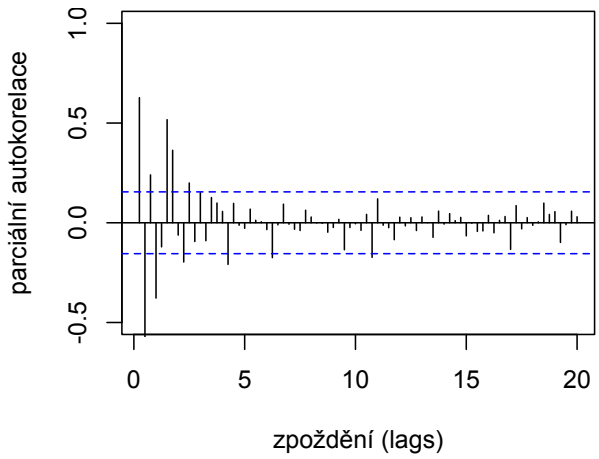
Pro logaritmy vrátil Dickey-Fullerův test hodnotu testové statistiky $DF = -3{,}814$ a hladinu významnosti $p = 0{,}020$. Tím zamítáme nulovou hypotézu o nestacionaritě řady ve prospěch její stacionarity.

3.3.3 Korelogram upravené sezónní časové řady

Na obrázcích 12 a 13 lze nahlédnou korelogram nad logaritmy sezónní časové řady.



Obrázek 12: Korelogram ACF logaritmů sezónní řady



Obrázek 13: Korelogram PACF logaritmů sezónní řady

Z korelogramu ACF můžeme nahlédnou, že bude třeba k vyhlazení „periodického“ trendu nejspíše proces klouzavých průměrů MA(q) vysokého řádu, odhadem $q \leq 15$. Naopak pro autoregresní složku AR(p) modelu SARIMA tedy bude nejspíše platit $p \leq 4$, což se lze dovtípit ze čtvrtého sloupečku korelace u meze v korelogramu PACF.

3.3.4 Určení nejvhodnějšího modelu

Protože se efekty složek AR(p) a MA(q) mohou v jednom modelu SARIMA navzájem „rušit“, nelze apriorně říci, který model typu SARIMA(i, 0, j)(P, D, Q) bude vhodný. Postupným diferencováním je třeba se nejdříve zbavit trendu (a určit d), poté je třeba dalším diferencováním odstranit sezónnost (a určit D). Pak zbývá určit řád AR(p) a MA(q), rovněž tak SAR(P) a SMA(Q). Situace může být nepřehledná, pomůže opět exhaustivní přístup, případně funkce `auto.arima()`, která automaticky sama najde model SARIMA s nejnižším možným AIC (Akaikeho informačním kritériem).

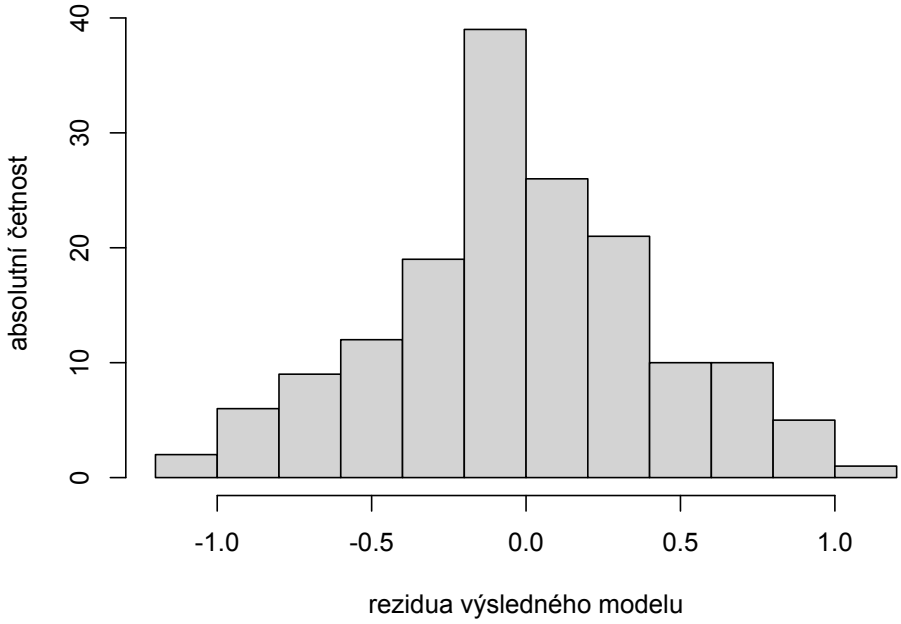
Pomocí funkce funkce `auto.arima()` bylo zjištěno, že AIC minimalizuje model je SARIMA(3, 1, 3)(1, 0, 0)$_4$. V tabulce 3 je sumář modelu SARIMA(3, 1, 3)(1, 0, 0)$_4$; v sumáři není obsažena konstanta, byla tedy automaticky procedurou vyloučena jako nesignifikantní.

	odhad	střední chyba	z-hodnota	p-hodnota
AR(1)	-0,820	0,070	-11,641	$<$ 0,001
AR(2)	-0,964	0,037	-26,253	$<$ 0,001
AR(3)	-0,746	0,067	-11,078	$<$ 0,001
MA(1)	1,036	0,096	10,844	$<$ 0,001
MA(2)	0,662	0,116	5,716	$<$ 0,001
MA(3)	-0,126	0,101	-1,237	0,216
SAR(1)	-0,870	0,041	-20,989	$<$ 0,001

Tabulka 3: Sumář modelu SARIMA(3, 1, 3)(1, 0, 0)$_4$

3.3.5 Diagnostika modelu

Breusch-Godfreyho test (LM = 0,070, df = 1, p = 0,792) nezamítá pro tento model nulovou hypotézu o absenci autokorelace.

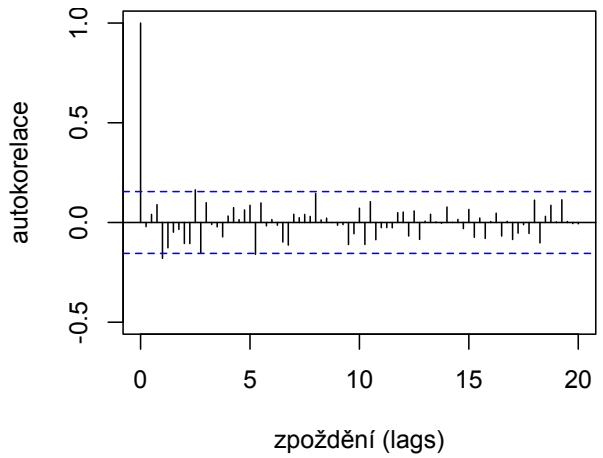


Obrázek 14: Histogram rozložení reziduí modelu SARIMA(3, 1, 3)(1, 0, 0)$_4$

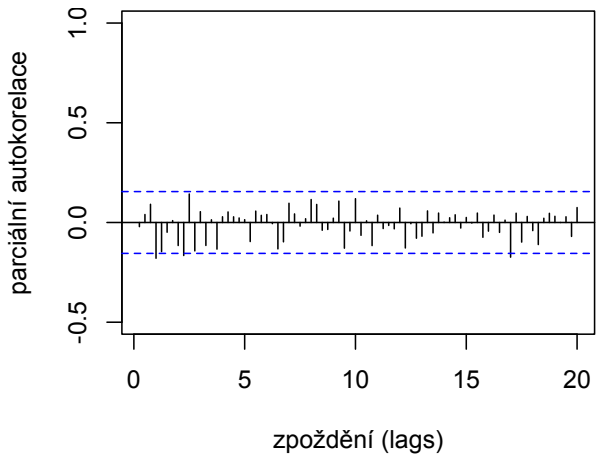
Shapiro-Wilkův test nezamítá nulovou hypotézu o normalitě reziduí ($W = 0{,}991$, $p = 0{,}367$) a Breusch-Paganův test nezamítá nulovou hypotézu o homoskedasticitě chybové složky ($BP = 0{,}445$, $df = 1$, $p = 0{,}505$).

Dále ještě obrázek 14 potvrzuje závěr Shapiro-Wilkova test, že rezidua modelu je možné považovat za normální.

Nakonec obrázky 15 a 16 naznačují chování reziduí modelu SARIMA$(3, 1, 3)(1, 0, 0)_4$. Korelace je v případě některých zpoždění hraničně překročena, což momentálně zanedbáme a budeme považovat za limitaci modelu.



Obrázek 15: Korelogram ACF reziduí modelu SARIMA$(3, 1, 3)(1, 0, 0)_4$ nad sezónní řadou



Obrázek 16: Korelogram PACF reziduí modelu SARIMA$(3, 1, 3)(1, 0, 0)_4$ nad sezónní řadou

3.3.6 Závěr

Pro sezónní časovou řadu byl zvolen jako nejvýstižnější model SARIMA$(3, 1, 3)(1, 0, 0)_4$ bez konstanty.

4 Apendix

Zde je uveden kód v jazyce R, ve kterém byly zpracovávány veškeré výpočty a rovněž generovány diagramy.

```
################################################################################
################################################################################
################################################################################

## instaluji a loaduji balíčky ------------------------------------------------

invisible(
    lapply(
        c(
            "xtable",
            "openxlsx",
            "tseries",
            "seasonal",
            "forecast",
            "lmtest",
            "car"
        ),
        function(my_package){

            if(!(my_package %in% rownames(installed.packages()))){

                install.packages(
                    my_package,
                    dependencies = TRUE,
                    repos = "http://cran.us.r-project.org"
                )

            }

            library(my_package, character.only = TRUE)

        }
    )
)


## ----------------------------------------------------------------------------

################################################################################

## nastavuji handling se zipováním v R ----------------------------------------

Sys.setenv(R_ZIPCMD = "C:/Rtools/bin/zip")
```

```
## ------------------------------------------------------------------------

################################################################################

## nastavuji pracovní složku -------------------------------------------------

while(!"__semestralni_prace__.R" %in% dir()){
    setwd(choose.dir())
}

mother_working_directory <- getwd()


## ------------------------------------------------------------------------

################################################################################

## vytvářím posložky pracovní složky -----------------------------------------

setwd(mother_working_directory)

for(my_subdirectory in c("vstupy", "vystupy")){

    if(!file.exists(my_subdirectory)){

        dir.create(file.path(

            mother_working_directory, my_subdirectory

        ))

    }

}


## ------------------------------------------------------------------------

################################################################################

## loaduji data --------------------------------------------------------------

setwd(
    paste(mother_working_directory, "vstupy", sep = "/")
)

time_series_1 <- read.table(

    file = "karcinom_tlusteho_streva.txt",
```

```
    header = TRUE,
    sep = ";",
    dec = ","

)

time_series_2 <- readLines(

    con = "ewcitmeas.dat",
    encoding = "UTF-8"

)


setwd(mother_working_directory)


## ------------------------------------------------------------------------

################################################################################

## (pre)processing dat ---------------------------------------------------------

#### upravuji druhou časovou řadu ----------------------------------------------

###### umazávám hashtag v headeru ----------------------------------------------

time_series_2 <- gsub("#", "", time_series_2)


###### nahrazuji hvězdičky chybějící hodnotou (NA) -----------------------------

#time_series_2 <- gsub("\\*", NA, time_series_2)


###### odmazávám iniciální mezeru, je-li přítomna ------------------------------

time_series_2 <- gsub("^\\s+", "", time_series_2)


###### všechny vícenásobné mezery měním na jednoduché a nahrazuji je
###### středníkem --------------------------------------------------------------

for(i in 1:length(time_series_2)){

    while(grepl("  ", time_series_2[i])){

        time_series_2[i] <- gsub("  ", " ", time_series_2[i])
```

```
    }

}

time_series_2 <- gsub(" ", ";", time_series_2)


###### vytvářím data.frame z "time_series_2" ---------------------------------

temp_time_series_2 <- NULL

for(i in 2:length(time_series_2)){

    temp_time_series_2 <- rbind(

        temp_time_series_2,
        strsplit(time_series_2[i], split = ";")[[1]]

    )

}


colnames(temp_time_series_2) <- strsplit(
    time_series_2[1], split = ";"
)[[1]]


temp_time_series_2 <- data.frame(

    temp_time_series_2,
    stringsAsFactors = FALSE

)


######## převádím dny, měsíce a roky na integery -----------------------------

for(i in 1:3){

    temp_time_series_2[, i] <- suppressWarnings(
        as.integer(
            temp_time_series_2[, i]
        )
    )

}
```

```
######## převádím ostatní hodnoty na reálná čísla ----------------------------

for(i in 4:dim(temp_time_series_2)[2]){

    temp_time_series_2[, i] <- suppressWarnings(
        as.numeric(
            temp_time_series_2[, i]
        )
    )

}


time_series_2 <- temp_time_series_2


###### agreguji data v "time_series_2" na měsíční řady ------------------------

temp_time_series_2 <- NULL

for(my_year in unique(time_series_2$YY)){

    my_data <- time_series_2[
        time_series_2[, "YY"] == my_year
        ,
    ]

    for(my_quarter in c(1, 2, 3, 4)){

        if(
            dim(
                my_data[
                    my_data[, "MM"] %in% c(
                        (3 * my_quarter - 2):(3 * my_quarter)
                    )
                    ,
                ]
            )[1] > 0
        ){

            temp_time_series_2 <- rbind(

                temp_time_series_2,
                c(

                    "year" = paste("19", my_year, sep = ""),
                    "quarter" = my_quarter,
                    apply(
                        my_data[
```

```
                            my_data[, "MM"] %in% c(
                                (3 * my_quarter - 2):(3 * my_quarter)
                            ),
                            setdiff(colnames(my_data), c("DD", "MM", "YY"))
                        ],
                        2,
                        sum,
                        na.rm = TRUE
                    )

                )

            )

        }

    }

}


time_series_2 <- data.frame(

    temp_time_series_2,
    stringsAsFactors = FALSE

)


for(i in 3:dim(time_series_2)[2]){

    time_series_2[, i] <- as.numeric(
        time_series_2[, i]
    )

}


## ------------------------------------------------------------------------

################################################################################

## převádím obě řady na objekt typu ts() -------------------------------------

ts_1 <- ts(

    time_series_1[, "incidence"],
    start = min(time_series_1[, "rok"]),
    end = max(time_series_1[, "rok"]),
```

```
    frequency = 1

)

ts_2 <- ts(

    time_series_2[, "London"],
    start = c(min(as.integer(time_series_2[, "year"])), 1),
    end = c(max(as.integer(time_series_2[, "year"])), 4),
    frequency = 4

)


## ------------------------------------------------------------------------

################################################################################

## vykresluji diagramy pro obě časové řady ----------------------------------

setwd(paste(mother_working_directory, "vystupy", sep = "/"))

for(i in c(1, 2)){

    cairo_ps(
        file = paste(
            "diagram_casova_rada_",
            i,
            ".eps",
            sep = ""
        ),
        width = 8,
        height = 4,
        pointsize = 12
    )

    par(mar = c(4.1, 4.1, 0.5, 0.1))

    plot(
        x = get(paste("ts_", i, sep = "")),
        xlab = "rok",
        ylab = if(
            i == 1
        ){
            "# nových případů / 100 000 osob"
        }else{
            "# případů celkově"
        },
        col = "blue"
```

```
    )

    dev.off()

}

setwd(mother_working_directory)


## ------------------------------------------------------------------------

################################################################################

## korelogramy -----------------------------------------------------------------

setwd(paste(mother_working_directory, "vystupy", sep = "/"))

for(my_correlogram_type in c("acf", "pacf")){

    for(i in c(1, 2)){

        cairo_ps(
            file = paste(
                "my_",
                my_correlogram_type,
                "_diagram_",
                i,
                ".eps",
                sep = ""
            ),
            width = 4,
            height = 3,
            pointsize = 12
        )

        par(mar = c(4.1, 4.1, 0.1, 0.1))

        do.call(
            what = my_correlogram_type,
            args = list(
                x = get(paste("ts_", i, sep = "")),
                xlab = "zpoždění (lags)",
                ylab = if(
                    my_correlogram_type == "acf"
                ){
                    "autokorelace"
                }else{
                    "parciální autokorelace"
                },
```

```
                main = "",
                ylim = c(-0.5, 1.0),
                lag.max = if(i == 1){20}else{80}
            )
        )

        dev.off()

    }

}

setwd(mother_working_directory)


## ------------------------------------------------------------------------

################################################################################

## řešení pro jednotlivé řady: ------------------------------------------------

## nesezónní řada --------------------------------------------------------------

#### Augmentovaný Dickey-Fullerův test ----------------------------------------

adf.test(ts_1)
adf.test(log(ts_1))     # nezamítáme H_0, takže nestacionarita
adf.test(diff(log(ts_1), diff = 1))
                        # zamítáme H_0, stacionarita

# Augmented Dickey-Fuller Test

# data:  diff(log(ts_1), diff = 1)
# Dickey-Fuller = -3.8287, Lag order = 3, p-value = 0.02878
# alternative hypothesis: stationary


##### ACF a PACF po diferenciaci a logaritmování řady -------------------------

acf(diff(log(ts_1), diff = 1), lag.max = 20)
pacf(diff(log(ts_1), diff = 1), lag.max = 20)


setwd(paste(mother_working_directory, "vystupy", sep = "/"))

for(my_correlogram_type in c("acf", "pacf")){

    cairo_ps(
        file = paste(
```

```
            "my_",
            my_correlogram_type,
            "_diagram_after_dif_1",
            ".eps",
            sep = ""
        ),
        width = 4,
        height = 3,
        pointsize = 12
    )

    par(mar = c(4.1, 4.1, 0.1, 0.1))

    do.call(
        what = my_correlogram_type,
        args = list(
            x = diff(log(ts_1), diff = 1),
            xlab = "zpoždění (lags)",
            ylab = if(
                my_correlogram_type == "acf"
            ){
                "autokorelace"
            }else{
                "parciální autokorelace"
            },
            main = "",
            ylim = c(-0.5, 1.0),
            lag.max = 20
        )
    )

    dev.off()

}

setwd(mother_working_directory)


#### vytvářím tabulku s možnými modely ----------------------------------------

d <- 1

my_table <- NULL

for(p in 0:4){

    for(q in 0:4){

        my_arima <- arima(
```

```
            log(ts_1),
            order = c(p, d, q)
        )

        my_table <- rbind(

            my_table,
            c(
                "p" = p,
                "d" = d,
                "q" = q,
                "AIC" = my_arima$aic,
                "ljung_box_p" = Box.test(
                    my_arima$residuals,
                    lag = 20,
                    type = "Ljung-Box",
                    fitdf = sum(c(p, d, q))
                )$p.value,
                "breusch_godfrey_p" = bgtest(
                    lm(residuals(my_arima) ~ 1)
                )$p.value,
                "shapiro_p" = shapiro.test(
                    my_arima$residuals
                )$p.value,
                "breusch_pagan_p" = unname(
                    bptest(
                        residuals ~ year,
                        data = data.frame(
                            "residuals" = c(residuals(my_arima)),
                            "year" = c(
                                1:length(residuals(my_arima))
                            )
                        )
                    )$p.value
                )
            )

        )

    }

}


print(
    xtable(
        my_table,
        align = rep("", ncol(my_table) + 1),
        digits = c(0, 0, 0, 0, 3, 3, 3, 3, 3)
```

```
    ),
    floating = FALSE,
    tabular.environment = "tabular",
    hline.after = NULL,
    include.rownames = TRUE,
    include.colnames = TRUE,
    format.args = list(decimal.mark = ",")
)


#### tisknu ještě výstupy diagnostických testů pro nejvhodnější model ---------

p <- 2
d <- 1
q <- 2


my_arima <- arima(
    log(ts_1),
    order = c(p, d, q)
)


###### Ljung-Boxův test ------------------------------------------------------

Box.test(
    arima(
        log(ts_1),
        order = c(p, d, q)
    )$residuals,
    lag = 20,
    type = "Ljung-Box",
    fitdf = sum(c(p, d, q))
)


###### Breusch-Godfreyho test ------------------------------------------------

bgtest(
    lm(residuals(my_arima) ~ 1)
)


###### Shapiro-Wilkův test ---------------------------------------------------

shapiro.test(
    arima(
        log(ts_1),
        order = c(p, d, q)
```

```
    )$residuals
)


###### Breusch-Paganův test ------------------------------------------------

bptest(
    residuals ~ guarter,
    data = data.frame(
        "residuals" = c(0, residuals(my_arima)),
        "guarter" = rep(1:4, 10)
    )
)


#### ukládám histogram reziduí ---------------------------------------------

setwd(paste(mother_working_directory, "vystupy", sep = "/"))

cairo_ps(
    file = paste(
        "my_histogram_1",
        ".eps",
        sep = ""
    ),
    width = 6,
    height = 4,
    pointsize = 12
)

par(mar = c(4.1, 4.1, 0.1, 0.1))

hist(
    x = residuals(my_arima),
    col = "lightgrey",
    xlim = c(-0.15, 0.15),
    xlab = "rezidua výsledného modelu",
    ylab = "absolutní četnost",
    main = "",
    ylim = c(0, 12)
)

lines(density(residuals(my_arima)), col = "red")

dev.off()

setwd(mother_working_directory)
```

```
#### ukládám ACF a PACF pro výsledný model ----------------------------------

my_arima <- arima(
    log(ts_1),
    order = c(p, d, q)
)


setwd(paste(mother_working_directory, "vystupy", sep = "/"))

for(my_correlogram_type in c("acf", "pacf")){

    cairo_ps(
        file = paste(
            "my_",
            my_correlogram_type,
            "_diagram_final_1",
            ".eps",
            sep = ""
        ),
        width = 4,
        height = 3,
        pointsize = 12
    )

    par(mar = c(4.1, 4.1, 0.1, 0.1))

    do.call(
        what = my_correlogram_type,
        args = list(
            x = my_arima$residuals,
            xlab = "zpoždění (lags)",
            ylab = if(
                my_correlogram_type == "acf"
            ){
                "autokorelace"
            }else{
                "parciální autokorelace"
            },
            main = "",
            ylim = c(-0.5, 1.0),
            lag.max = 20
        )
    )

    dev.off()

}
```

```r
setwd(mother_working_directory)


#### tisknu výstup modelu ----------------------------------------------------

my_table <- unclass(coeftest(my_arima))

print(
    xtable(
        my_table,
        align = rep("", ncol(my_table) + 1),
        digits = c(0, 3, 3, 3, 3)
    ),
    floating = FALSE,
    tabular.environment = "tabular",
    hline.after = NULL,
    include.rownames = TRUE,
    include.colnames = TRUE,
    format.args = list(decimal.mark = ",")
)


## ----------------------------------------------------------------------------

################################################################################

## řešení pro jednotlivé řady: ------------------------------------------------

## sezónní řada ----------------------------------------------------------------

#### Augmentovaný Dickey-Fullerův test ---------------------------------------

adf.test(log(ts_2))     # zamítáme H_0, stacionarita

# Augmented Dickey-Fuller Test

# data:  log(ts_2)
# Dickey-Fuller = -3.8143, Lag order = 5, p-value = 0.0201
# alternative hypothesis: stationary


##### ACF a PACF po logaritmování řady ---------------------------------------

acf(diff(log(ts_2), lag = 4, differences = 1), lag.max = 80)
pacf(log(ts_2), lag.max = 80)


setwd(paste(mother_working_directory, "vystupy", sep = "/"))
```

```
for(my_correlogram_type in c("acf", "pacf")){

    cairo_ps(
        file = paste(
            "my_",
            my_correlogram_type,
            "_diagram_after_dif_2",
            ".eps",
            sep = ""
        ),
        width = 4,
        height = 3,
        pointsize = 12
    )

    par(mar = c(4.1, 4.1, 0.1, 0.1))

    do.call(
        what = my_correlogram_type,
        args = list(
            x = log(ts_2),
            xlab = "zpoždění (lags)",
            ylab = if(
                my_correlogram_type == "acf"
            ){
                "autokorelace"
            }else{
                "parciální autokorelace"
            },
            main = "",
            ylim = c(-0.5, 1.0),
            lag.max = 80
        )
    )

    dev.off()

}

setwd(mother_working_directory)


#### hledám nejlepší model ----------------------------------------------------

my_sarima <- arima(
    log(ts_2),
    order = c(3, 1, 3),
    seasonal = list(order = c(1, 0, 0), period = NA)
)
```

```
coeftest(my_sarima)

acf(my_sarima$resid)
pacf(my_sarima$resid)


print(
    xtable(
        my_table,
        align = rep("", ncol(my_table) + 1),
        digits = c(0, 0, 0, 0, 3, 3, 3, 3, 3)
    ),
    floating = FALSE,
    tabular.environment = "tabular",
    hline.after = NULL,
    include.rownames = TRUE,
    include.colnames = TRUE,
    format.args = list(decimal.mark = ",")
)


#### tisknu ještě výstupy diagnostických testů pro nejvhodnější model ---------

###### Ljung-Boxův test ----------------------------------------------------

Box.test(
    my_sarima$residuals,
    lag = 20,
    type = "Ljung-Box",
    fitdf = sum(c(p, d, q))
)


###### Breusch-Godfreyho test ----------------------------------------------

bgtest(
    lm(residuals(my_sarima) ~ 1)
)


###### Shapiro-Wilkův test -------------------------------------------------

shapiro.test(
    my_sarima$residuals
)


###### Breusch-Paganův test ------------------------------------------------
```

```
bptest(
    residuals ~ guarter,
    data = data.frame(
        "residuals" = c(residuals(my_sarima)),
        "guarter" = rep(1:4, 40)
    )
)


#### ukládám histogram reziduí ------------------------------------------------

setwd(paste(mother_working_directory, "vystupy", sep = "/"))

cairo_ps(
    file = paste(
        "my_histogram_2",
        ".eps",
        sep = ""
    ),
    width = 6,
    height = 4,
    pointsize = 12
)

par(mar = c(4.1, 4.1, 0.3, 0.1))

hist(
    x = c(residuals(my_sarima)),
    col = "lightgrey",
    xlim = c(-1.2, 1.2),
    xlab = "rezidua výsledného modelu",
    ylab = "absolutní četnost",
    main = ""
)

dev.off()

setwd(mother_working_directory)


#### ukládám ACF a PACF pro výsledný model ------------------------------------

setwd(paste(mother_working_directory, "vystupy", sep = "/"))

for(my_correlogram_type in c("acf", "pacf")){

    cairo_ps(
        file = paste(
            "my_",
```

```
            my_correlogram_type,
            "_diagram_final_2",
            ".eps",
            sep = ""
        ),
        width = 4,
        height = 3,
        pointsize = 12
    )

    par(mar = c(4.1, 4.1, 0.1, 0.1))

    do.call(
        what = my_correlogram_type,
        args = list(
            x = my_sarima$residuals,
            xlab = "zpoždění (lags)",
            ylab = if(
                my_correlogram_type == "acf"
            ){
                "autokorelace"
            }else{
                "parciální autokorelace"
            },
            main = "",
            ylim = c(-0.5, 1.0),
            lag.max = 80
        )
    )

    dev.off()

}

setwd(mother_working_directory)


#### tisknu výstup modelu ----------------------------------------------------

my_table <- unclass(coeftest(my_sarima))

print(
    xtable(
        my_table,
        align = rep("", ncol(my_table) + 1),
        digits = c(0, 3, 3, 3, 3)
    ),
    floating = FALSE,
    tabular.environment = "tabular",
```

```
    hline.after = NULL,
    include.rownames = TRUE,
    include.colnames = TRUE,
    format.args = list(decimal.mark = ",")
)


## ------------------------------------------------------------------------

###############################################################################
###############################################################################
###############################################################################
```

5 Reference

[1] R CORE TEAM. *R: A Language and Environment for Statistical Computing* [online]. Vienna, Austria: R Foundation for Statistical Computing, 2016. Dostupné z: https://www.R-project.org/

[2] MONSELL, Brian a Chris BLAKELY. *X-13ARIMA-SEATS and iMetrica*

[3] PRIESTLEY, M. B. *Non-linear and non-stationary time series analysis*. London San Diego: Academic Press, 1988. ISBN 0-12-564911-8.

[4] SAID, SAID E. a DAVID A. DICKEY. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika* [online]. 1984, **71**(3), 599–607. Dostupné z: doi:10.1093/biomet/71.3.599

[5] KWIATKOWSKI, Denis, Peter C.B. PHILLIPS, Peter SCHMIDT a Yongcheol SHIN. Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics* [online]. 1992, **54**(1-3), 159–178. Dostupné z: doi:10.1016/0304-4076(92)90104-y

[6] BROCKWELL, Peter J. a Richard A. DAVIS, ed. *Introduction to Time Series and Forecasting* [online]. B.m.: Springer New York, 2002. Dostupné z: doi:10.1007/b97391

[7] BREUSCH, T. S. TESTING FOR AUTOCORRELATION IN DYNAMIC LINEAR MODELS. *Australian Economic Papers* [online]. 1978, **17**(31), 334–355. Dostupné z: doi:10.1111/j.1467-8454.1978.tb00635.x

[8] ROYSTON, Patrick. An extension of Shapiro and Wilk's W test for normality to large samples. *Applied Statistics*. 1982, **4**, 115–124.

[9] BREUSCH, Trevor a Adrian PAGAN. A Simple Test for Heteroscedasticity and Random Coefficient Variation. *Econometrica* [online]. 1979, **47**(5), 1287–94. Dostupné z: https://EconPapers.repec.org/RePEc:ecm:emetrp:v:47:y:1979:i:5:p:1287-94

[10] TROTTIER, Helen, Pierre PHILIPPE a Roch ROY. *Emerging Themes in Epidemiology* [online]. 2006, **3**(1), 9. Dostupné z: doi:10.1186/1742-7622-3-9